ยุคลอาจ ชาญพานิชกิจการ

SOFT MAIL

ใช้ Visual Basic เขียนโปรแกรม CGI

การใช้ Visual Basic เขียน CGI ประโยชน์สำหรับท่านที่มีเวบเซอร์ฟเวอร์

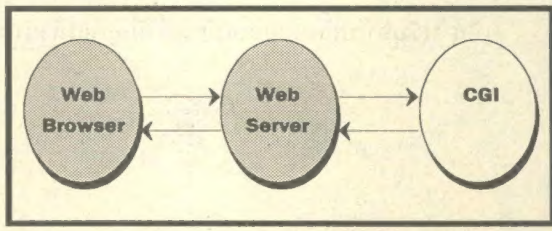
สวัสดีครับท่านผู้อ่านทุกท่าน เป็นอย่างไรบ้างครับกับเศรษฐกิจของประเทศ อย่างไรก็อดทนกันหน่อยนะครับ มาเข้าเรื่องของเราดีกว่า จากที่ผ่านมาไม่นานนี้ผมได้เขียนเรื่องเกี่ยวกับ Clipper กับ CGI ไป (ใน Internet Magazine) ซึ่งก็คงจะผ่านสายตาท่านผู้อ่านบ้างนะครับ มาถึงอีกเรื่องที่น่าจะนำมากล่าวถึงก็คือ การใช้ Visual Basic เขียน CGI (Common Gateway Interface) สิ่งนี้ผมคิดว่าคงจะเป็นประโยชน์กับผู้อ่านหลาย ๆ ท่านที่มีเวบเซอร์ฟเวอร์ที่วิ่งอยู่บนวินโดวส์ 3.1 วินโดวส์ 95 หรือวินโดวส์เอ็นที ที่ว่ามีประโยชน์ก็เพราะว่าคุณสามารถจะนำเอาความสามารถของ DBEngine มาใช้ในการทำงานติดต่อกับ Database หลาย ๆ ผลิตภัณฑ์ได้ CGI ที่ใช้ Visual Basic ในการเขียนนั้น ถูกสร้างขึ้นมาจาก Mr. Bob Denny ซึ่งเวบเซอร์ฟเวอร์ที่ Mr. Bob Denny สร้างขึ้นมาเพื่อสนับสนุน WinCGI นี้มีชื่อว่า WebSite ซึ่งคุณสามารถดาวน์โหลดได้ที่

```
http://www.city.net/win-httpd/
```

โดยที่เวบเซอร์ฟเวอร์นี้สามารถทำงานบนวินโดวส์ 3.X วินโดวส์ 95 และวินโดวส์เอ็นทีได้

เข้าใจการทำงานของ WinCGI

การทำงานของ WinCGI นี้จะมีตัวแปรต่าง ๆ ตามมาตรฐานของ CGI ทุกประการ และมีตัวแปรต่าง ๆ เพิ่มเติมขึ้นมาอีกด้วย ซึ่งผมจะข้ามไป เพราะว่าคุณสามารถที่จะหาเอกสารประกอบเกี่ยวกับตัวแปรต่าง ๆ เหล่านี้ได้อยู่แล้ว แต่ผมจะกล่าวถึงสิ่งที่เรียกว่า มันทำงานอย่างไร ส่งข้อมูลจากเวบเซอร์ฟเวอร์มาได้อย่างไร ตามมาตรฐานการทำงานของ CGI นั้นจะเป็นดังรูปที่ 1 กล่าวคือเวบเบราเซอร์ส่งข้อมูลมายังเวบเซอร์ฟเวอร์ และจะส่งต่อไปให้ CGI โดยเวบเซอร์ฟเวอร์ เมื่อ CGI ทำงานประมวลผลตามข้อมูลที่ให้มาแล้วจะส่งผลลัพธ์กลับให้เวบเซอร์ฟเวอร์ ซึ่งจะทำการส่งต่อให้กับเวบเบราเซอร์อีกทีเพื่อนำไปแสดงให้ผู้ใช้ได้ดู



รูปที่ 1 แสดงการทำงานของ CGI กับเวบเซอร์ฟเวอร์

คราวนี้เราจะมาดูกันว่าข้อมูลที่ส่งมาจากเวบเซอร์ฟเวอร์สู่ CGI นี้ถูกส่งมาอย่างไร ใน WinCGI นี้จะมีวิธีการส่งอยู่ 2 แบบ เหมือนกับการใช้ CGI-BIN คือ POST กับ GET ความหมายของ POST กับ GET ใน WinCGI (CGI-WIN) กับ CGI-BIN จะมีวิธีที่แตกต่างกันพอสมควร ดังนี้ ถ้าคุณใช้วิธีส่งโดยใช้ POST ใน CGI-BIN ตัวเวบเซอร์ฟเวอร์จะใช้ STDIN (Standard Input) ในการส่งข้อมูลให้กับ CGI โปรแกรม ซึ่งต้องเขียนโปรแกรมไปอ่านเข้ามาเอาเอง แต่ถ้าเป็น WinCGI (ซึ่งต่อไปนี้ผมจะเรียกว่า CGI-WIN แทน) แล้วจะสร้าง Temporary File ขึ้นมาและบันทึกข้อมูลฟิลด์ในฟอร์มลงไปเก็บเอาไว้ ส่วนการส่งโดยวิธี GET นั้น CGI-BIN และ CGI-WIN จะมีการทำงานที่เหมือนกันคือ ส่งต่อถ่าย url ไป

```
http://127.0.0.1/cgi-bin/
readpara.exe?name=Yukonart&Lname=Chanpanichkijjakarn&BDATE=
27-02-71&SEX=male
```

หลังจากที่คุณเข้าใจแล้วว่ามันทำงานอย่างไร จุดต่อไปก็คือเราจะหาค่าต่าง ๆ เหล่านี้ได้อย่างไร ไม่ยากครับ ส่วนตัวผมเองมองว่า CGI-WIN นี้มีวิธีการส่งข้อมูลที่ง่ายต่อการนำมาใช้ เพราะว่าข้อมูลจะถูกสร้างขึ้นมาเก็บไว้ใน Temp ไฟล์ ถ้าคุณอยากจะหาค่าอะไรก็มาเปิดเอาจาก Temp ไฟล์นี้ ขอบอกก่อนว่า CGI-WIN นี้คุณจะใช้โปรแกรมภาษาอะไรก็ได้เขียนขึ้นมาไม่ว่าจะเป็นบนดอสหรือวินโดวส์ แต่ขอให้ปฏิบัติตามเงื่อนไขเหล่านี้ ที่กล่าวเช่นนี้เพราะว่าผมได้ลองดูแล้ว กลับมาที่ CGI-WIN ต่อ เมื่อคุณ Submit ที่ฟอร์มใน Page แล้ว ข้อมูลจะถูกส่งเข้ามายังเวบเซอร์ฟเวอร์ ตัวเวบเซอร์ฟเวอร์เองจะเรียกโปรแกรม CGI ขึ้นมาทำงาน โดยส่งชื่อไฟล์ Temporary ไปให้โปรแกรม CGI ซึ่งใน Temp ไฟล์นี้จะเก็บค่าตัวแปรต่าง ๆ รวมชื่อไฟล์ Content และ Output File ด้วย พูดง่าย ๆ ว่า เมื่อมีการ Submit แล้วเวบเซอร์ฟเวอร์จะทำการสร้างไฟล์ขึ้นมาให้ 3 ไฟล์ ไฟล์แรกเป็นไฟล์ที่เก็บค่าตัวแปร Env ต่าง ๆ ไว้ ผมจะขอเรียกว่าเป็น CGI Profile ไฟล์ที่สองเป็น Content ไฟล์ Content ไฟล์นี้ก็คือข้อมูลบนฟอร์มที่ผู้ใช้ได้กรอกเอาไว้นั่นเอง แต่ไฟล์ที่สองจะเกิดก็ต่อเมื่อคุณสั่งให้ส่งโดยวิธี POST เท่านั้นนะครับ ถ้าคุณสั่งโดยวิธี GET จะไม่มี แต่ข้อมูลจะไปอยู่ในตัวแปร QueryString แทน ดูตารางที่ 1

ตารางที่ 1

วิธี	ส่งโดย
POST	ข้อมูลถูกสร้างไว้ใน Content File และขนาดของไฟล์คุณจะหาได้จากตัวแปร ContentLength
GET	ข้อมูลถูกส่งมาอยู่ในตัวแปร QueryString

หมายเหตุ : ข้อมูลจะถูกแบ่งโดยแต่ละฟิลด์จะขั้นด้วยตัว "&" ดังนี้
Yukonart&Lname=Chanpanichkijjakarn&BDATE=27-02-71&SEX=male

ส่วนไฟล์ที่สามนั้นจะเป็น Output file ซึ่งเป็นไฟล์ที่ทำหน้าที่รับข้อมูลที่ต้องการส่งให้กับเวบเซอร์ฟเวอร์ไปแจ้งแก่ผู้ใช้

ถ้าจะไม่ให้ดูตัวอย่างไฟล์ก็คงจะใจร้ายไปนิด ผมจึงนำตัวอย่างไฟล์ CGI Profile ที่เก็บค่าต่าง ๆ มาให้ดูเป็นตัวอย่างที่ 1 ไฟล์นี้ถูกสร้างโดยเวบเซอร์ฟเวอร์เมื่อผู้ใช้กดปุ่ม Submit มานะครับ

ตัวอย่างที่ 1 ไฟล์ CGI Profile

```
[CGI]
Request Protocol=HTTP/1.0
Request Method=POST
Executable Path=/cgi-win/checktm.exe
Server Software=WinHttpd/1.4b (Shareware Non-Commercial
License)
Server Name=192.1.1.200
Server Port=80
Server Admin=aong1@oeic-hp
CGI Version=CGI/1.1 WIN
Remote Address=192.1.1.111
Content Type=application/x-www-form-urlencoded
Content Length=44

[System]
Debug Mode=No
Output File=F:\PUBLIC\WEB\TEMP\HS00228A.OUT
Content File=F:\PUBLIC\WEB\TEMP\HS00228A.INP

[Form Literal]
TELCODE=23432432
bDate=
eDate=
Submit=Submit

[Accept]
image/gif=Yes
image/x-xbitmap=Yes
image/jpeg=Yes
image/pjpeg=Yes
*/*=Yes

[Extra Headers]
Referer=http://192.1.1.177/ct.htm
Accept-Language=th, en
UA-pixels=640x480
UA-color=color16
UA-OS=Windows 95
UA-CPU=x86
User-Agent=Mozilla/2.0 (compatible; MSIE 3.0; Windows 95)
Host=192.1.1.177
Connection=Keep-Alive
Pragma=No-Cache
```

คุณจะเห็นว่ามีการแบ่งข้อมูลออกเป็น Section ต่าง ๆ ตามอย่าง INI ไฟล์ของวินโดวส์ ส่วนจุดที่สำคัญ ๆ จะดีดังนี้ครับ

```
Request Method=POST
```

ตัวแปรนี้บอกวิธีที่เวบเซอร์ฟเวอร์ส่งข้อมูลมาให้กับ CGI โปรแกรม

```
Content Length=44
```

ตัวแปรนี้บอกจำนวนความยาวของข้อมูลที่ส่งมาให้ อีกนัยหนึ่งก็คือขนาดของไฟล์ Content ซึ่งคุณสามารถที่จะนำขนาดมาเปิดไฟล์อ่านได้เลย

```
Output File=F:\PUBLIC\WEB\TEMP\HS00228A.OUT
Content File=F:\PUBLIC\WEB\TEMP\HS00228A.INP
```

ตัวแปรสองตัวนี้ ตัวแรกบอกไฟล์ Output ตัวที่สองบอกไฟล์ Content ที่ผมได้กล่าวถึงไว้ ซึ่งเก็บข้อมูลในฟอร์มที่ถูกส่งมาจากเวบเซอร์ฟเวอร์ เมื่อคุณรู้วิธีการรับข้อมูลจากเวบเซอร์ฟเวอร์ และการอ่านค่าตัวแปรต่าง ๆ รวมถึงข้อมูลในฟอร์มแล้ว จุดต่อที่คุณควรจะรู้ก็คือ การส่งข้อมูลกลับไปยังเวบเซอร์ฟเวอร์ ซึ่งก็ไม่ยากอีกนั่นแหละ ให้คุณดู Output File ตัวแปรนี้จะบอกชื่อไฟล์ Output ที่ไฟล์ Output นี้ให้คุณเขียนข้อมูลลงไปในไฟล์ในรูปแบบของ HTML และเจ้าเวบเซอร์ฟเวอร์จะเปิดขึ้นมาอ่านเองครับ เมื่อคุณทำความเข้าใจเป็นที่เรียบร้อยแล้วต่อไปเราจะมาดูกันว่า Visual Basic จะใช้งานได้อย่างไร เริ่มจากคุณต้องไปหาเวบเซอร์ฟเวอร์ที่ชื่อเวบไซท์ หรืออะไรก็ได้ที่สามารถรับการทำงานในแบบ CGI-WIN ได้ แน่นอนว่ามันจะต้องเป็นเวบเซอร์ฟเวอร์ที่วิ่งอยู่บนวินโดวส์แน่นอน อย่างที่สองคือ ไฟล์ CGI32.BAS หรือ CGI.BAS ซึ่งเป็น Library Module ที่คุณต้องใช้ในการเขียน CGI คุณสามารถไปดาวน์โหลดมาได้ที่ผมบอกไว้ในช่วงแรก ๆ หรือไม่ก็พิมพ์ตามลิสติงที่ 1 นี้และบันทึกไว้เป็นชื่อ CGI32.BAS

ลิสติงที่ 1 ไฟล์ CGI32.BAS

```
Attribute VB_Name = "CGI_Framework"
'---------------------------------------
' VERSION: 1.7 (December 3, 1995)
' AUTHOR: Robert B. Denny <rdenny@netcom.com>
Option Explicit
' ===================
' Manifest Constants
' ===================
Const MAX_CMDARGS = 8 ' Max # of command line args
Const ENUM_BUF_SIZE = 4096 ' Key enumeration buffer, see GetProfile()
' These are the limits in the server
Const MAX_XHDR = 100 ' Max # of "extra" request headers
Const MAX_ACCTYPE = 100 ' Max # of Accept: types in request
Const MAX_FORM_TUPLES = 100 ' Max # form key=value pairs
Const MAX_HUGE_TUPLES = 16 ' Max # "huge" form fields
Const MAX_FILE_TUPLES = 16 ' Max # of uploaded file tuples
' =====
' Types
' =====
Type Tuple ' Used for Accept: and "extra" headers
 key As String ' and for holding POST form key=value pairs
 value As String
End Type
Type FileTuple ' Used for form-based file uploads
 key As String ' Form field name
 file As String ' Local tempfile containing uploaded file
 length As Long ' Length in bytes of uploaded file
 type As String ' Content type of uploaded file
 encoding As String ' Content-transfer encoding of uploaded file
 name As String ' Original name of uploaded file
End Type
Type HugeTuple ' Used for "huge" form fields
 key As String ' Keyword (decoded)
 offset As Long ' Byte offset into Content File of value
 length As Long ' Length of value, bytes
End Type
' ================
' Global Constants
' ================
'
' -----
' Error Codes
' -----
'
Global Const ERR_ARGCOUNT = 32767
Global Const ERR_BAD_REQUEST = 32766 ' HTTP 400
Global Const ERR_UNAUTHORIZED = 32765 ' HTTP 401
Global Const ERR_PAYMENT_REQUIRED = 32764 ' HTTP 402
Global Const ERR_FORBIDDEN = 32763 ' HTTP 403
Global Const ERR_NOT_FOUND = 32762 ' HTTP 404
Global Const ERR_INTERNAL_ERROR = 32761 ' HTTP 500
Global Const ERR_NOT_IMPLEMENTED = 32760 ' HTTP 501
Global Const ERR_TOO_BUSY = 32758 ' HTTP 503 (experimental)
```

→

ลิสติงที่ 1 (ต่อ)

```
Global Const ERR_NO_FIELD = 32757 ' GetxxxField "no field"
Global Const CGI_ERR_START = 32757 ' Start of our errors

' ====================
' CGI Global Variables
' ====================
'
' ------------
' Standard CGI variables
' ------------
'
Global CGI_ServerSoftware As String
Global CGI_ServerName As String
Global CGI_ServerPort As Integer
Global CGI_RequestProtocol As String
Global CGI_ServerAdmin As String
Global CGI_Version As String
Global CGI_RequestMethod As String
Global CGI_RequestKeepAlive As Integer
Global CGI_LogicalPath As String
Global CGI_PhysicalPath As String
Global CGI_ExecutablePath As String
Global CGI_QueryString As String
Global CGI_RequestRange As String
Global CGI_Referer As String
Global CGI_From As String
Global CGI_UserAgent As String
Global CGI_RemoteHost As String
Global CGI_RemoteAddr As String
Global CGI_AuthUser As String
Global CGI_AuthPass As String
Global CGI_AuthType As String
Global CGI_AuthRealm As String
Global CGI_ContentType As String
Global CGI_ContentLength As Long
'
' ----------
' HTTP Header Arrays
' ----------
'
Global CGI_AcceptTypes(MAX_ACCTYPE) As Tuple ' Accept: types
Global CGI_NumAcceptTypes As Integer ' # of live entries in array
Global CGI_ExtraHeaders(MAX_XHDR) As Tuple ' "Extra" headers
Global CGI_NumExtraHeaders As Integer ' # of live entries in array
'
' --------
' POST Form Data
' --------
'
Global CGI_FormTuples(MAX_FORM_TUPLES) As Tuple ' POST form key=value pairs
Global CGI_NumFormTuples As Integer ' # of live entries in array
Global CGI_HugeTuples(MAX_HUGE_TUPLES) As HugeTuple ' Form "huge tuples
Global CGI_NumHugeTuples As Integer ' # of live entries in array
Global CGI_FileTuples(MAX_FILE_TUPLES) As FileTuple ' File upload tuples
Global CGI_NumFileTuples As Integer ' # of live entries in array
'
' ---------
' System Variables
' ---------
'
Global CGI_GMTOffset As Variant ' GMT offset (time serial)
Global CGI_ContentFile As String ' Content/Input file pathname
Global CGI_OutputFile As String ' Output file pathname
Global CGI_DebugMode As Integer ' Script Tracing flag from server
'
' ========================
' Windows API Declarations
' ========================
'
' NOTE: Declaration of GetPrivateProfileString is specially done to
' permit enumeration of keys by passing NULL key value. See GetProfile().
' Both the 16-bit and 32-bit flavors are given below. We DO NOT
' recommend using 16-bit VB4 with WebSite!
'
#If Win32 Then
Declare Function GetPrivateProfileString Lib "kernel32" _
```

ลิสติงที่ 1 (ต่อ)

```
 Alias "GetPrivateProfileStringA" _
 (ByVal lpApplicationName As String, _
 ByVal lpKeyName As Any, _
 ByVal lpDefault As String, _
 ByVal lpReturnedString As String, _
 ByVal nSize As Long, _
 ByVal lpFileName As String) As Long
#Else
Declare Function GetPrivateProfileString Lib "Kernel" _
 (ByVal lpSection As String, _
 ByVal lpKeyName As Any, _
 ByVal lpDefault As String, _
 ByVal lpReturnedString As String, _
 ByVal nSize As Integer, _
 ByVal lpFileName As String) As Integer
#End If
'
'
' ===============
' Local Variables
' ===============
'
Dim CGI_ProfileFile As String ' Profile file pathname
Dim CGI_OutputFN As Integer ' Output file number
Dim ErrorString As String

'_____________________________________
'
' Return True/False depending on whether a form field is present.
' Typically used to detect if a checkbox in a form is checked or
' not. Unchecked checkboxes are omitted from the form content.
'
'_____________________________________
Function FieldPresent(key As String) As Integer
 Dim i As Integer

 FieldPresent = False ' Assume failure

 For i = 0 To (CGI_NumFormTuples - 1)
 If CGI_FormTuples(i).key = key Then
 FieldPresent = True ' Found it
 Exit Function ' ** DONE **
 End If
 Next i
 ' Exit with FieldPresent still False
End Function


'_____________________________________
'
' ErrorHandler() - Global error handler
'
' If a VB runtime error occurs dusing execution of the program, this
' procedure generates an HTTP/1.0 HTML-formatted error message into
' the output file, then exits the program.
'
' This should be armed immediately on entry to the program's main()
' procedure. Any errors that occur in the program are caught, and
' an HTTP/1.0 error messsage is generated into the output file. The
' presence of the HTTP/1.0 on the first line of the output file causes
' NCSA httpd for WIndows to send the output file to the client with no
' interpretation or other header parsing.
'_____________________________________
Sub ErrorHandler(code As Integer)

 Seek #CGI_OutputFN, 1 ' Rewind output file just in case
 Send ("HTTP/1.0 500 Internal Error")
 Send ("Server: " + CGI_ServerSoftware)
 Send ("Date: " + WebDate(Now))
 Send ("Content-type: text/html")
 Send ("")
 Send ("<HTML><HEAD>")
 Send ("<TITLE>Error in " + CGI_ExecutablePath + "</TITLE>")
 Send ("</HEAD><BODY>")
 Send ("<H1>Error in " + CGI_ExecutablePath + "</H1>")
 Send ("An internal Visual Basic error has occurred in " + CGI_ExecutablePath + ".")
 Send ("<PRE>" + ErrorString + "</PRE>")
```

ลิสติงที่ 1 (ต่อ)

```
 Send ("<I>Please</I> note what you were doing when this problem occurred,")
 Send ("so we can identify and correct it. Write down the Web page you were using,")
 Send ("any data you may have entered into a form or search box, and")
 Send ("anything else that may help us duplicate the problem. Then contact the")
 Send ("administrator of this service: ")
 Send ("<A HREF=""mailto:" & CGI_ServerAdmin & """>")
 Send ("<ADDRESS>&lt;" + CGI_ServerAdmin + "&gt;</ADDRESS>")
 Send ("</A></BODY></HTML>")

 Close #CGI_OutputFN

 '======
 End ' Terminate the program
 '======
End Sub

'-------------------------------------
'
' GetAcceptTypes() - Create the array of accept type structs
'
' Enumerate the keys in the [Accept] section of the profile file,
' then get the value for each of the keys.
'-------------------------------------
Private Sub GetAcceptTypes()
 Dim sList As String
 Dim i As Integer, j As Integer, l As Integer, n As Integer

 sList = GetProfile("Accept", "") ' Get key list
 l = Len(sList) ' Length incl. trailing null
 i = 1 ' Start at 1st character
 n = 0 ' Index in array
 Do While ((i < l) And (n < MAX_ACCTYPE)) ' Safety stop here
 j = InStr(i, sList, Chr$(0)) ' J -> next null
 CGI_AcceptTypes(n).key = Mid$(sList, i, j - i) ' Get Key, then value
 CGI_AcceptTypes(n).value = GetProfile("Accept", CGI_AcceptTypes(n).key)
 i = j + 1 ' Bump pointer
 n = n + 1 ' Bump array index
 Loop
 CGI_NumAcceptTypes = n ' Fill in global count

End Sub

'-------------------------------------
'
' GetArgs() - Parse the command line
'
' Chop up the command line, fill in the argument vector, return the
' argument count (similar to the Unix/C argc/argv handling)
'-------------------------------------
Private Function GetArgs(argv() As String) As Integer
 Dim buf As String
 Dim i As Integer, j As Integer, l As Integer, n As Integer

 buf = Trim$(Command$) ' Get command line

 l = Len(buf) ' Length of command line
 If l = 0 Then ' If empty
 GetArgs = 0 ' Return argc = 0
 Exit Function
 End If

 i = 1 ' Start at 1st character
 n = 0 ' Index in argvec
 Do While ((i < l) And (n < MAX_CMDARGS)) ' Safety stop here
 j = InStr(i, buf, " ") ' J -> next space
 If j = 0 Then Exit Do ' Exit loop on last arg
 argv(n) = Trim$(Mid$(buf, i, j - i)) ' Get this token, trim it
 i = j + 1 ' Skip that blank
 Do While Mid$(buf, i, 1) = " " ' Skip any additional whitespace
 i = i + 1
 Loop
 n = n + 1 ' Bump array index
 Loop

 argv(n) = Trim$(Mid$(buf, i, (l - i + 1))) ' Get last arg
 GetArgs = n + 1 ' Return arg count
```

→

ลิสติงที่ 1 (ต่อ)

```
End Function

'_____________________________________
'
' GetExtraHeaders() - Create the array of extra header structs
'
' Enumerate the keys in the [Extra Headers] section of the profile file,
' then get the value for each of the keys.
'_____________________________________
Private Sub GetExtraHeaders()
 Dim sList As String
 Dim i As Integer, j As Integer, l As Integer, n As Integer

 sList = GetProfile("Extra Headers", "") ' Get key list
 l = Len(sList) ' Length incl. trailing null
 i = 1 ' Start at 1st character
 n = 0 ' Index in array
 Do While ((i < l) And (n < MAX_XHDR)) ' Safety stop here
 j = InStr(i, sList, Chr$(0)) ' J -> next null
 CGI_ExtraHeaders(n).key = Mid$(sList, i, j - i) ' Get Key, then value
 CGI_ExtraHeaders(n).value = GetProfile("Extra Headers", CGI_ExtraHeaders(n).key)
 i = j + 1 ' Bump pointer
 n = n + 1 ' Bump array index
 Loop
 CGI_NumExtraHeaders = n ' Fill in global count

End Sub

'_____________________________________
'
' GetFormTuples() - Create the array of POST form input key=value pairs
'
'_____________________________________
Private Sub GetFormTuples()
 Dim sList As String
 Dim i As Integer, j As Integer, k As Integer
 Dim l As Integer, m As Integer, n As Integer
 Dim s As Long
 Dim buf As String
 Dim extName As String
 Dim extFile As Integer
 Dim extlen As Long

 n = 0 ' Index in array

 '
 ' Do the easy one first: [Form Literal]
 '
 sList = GetProfile("Form Literal", "") ' Get key list
 l = Len(sList) ' Length incl. trailing null
 i = 1 ' Start at 1st character
 Do While ((i < l) And (n < MAX_FORM_TUPLES)) ' Safety stop here
 j = InStr(i, sList, Chr$(0)) ' J -> next null
 CGI_FormTuples(n).key = Mid$(sList, i, j - i) ' Get Key, then value
 CGI_FormTuples(n).value = GetProfile("Form Literal", CGI_FormTuples(n).key)
 i = j + 1 ' Bump pointer
 n = n + 1 ' Bump array index
 Loop
 '
 ' Now do the external ones: [Form External]
 '
 sList = GetProfile("Form External", "") ' Get key list
 l = Len(sList) ' Length incl. trailing null
 i = 1 ' Start at 1st character
 extFile = FreeFile
 Do While ((i < l) And (n < MAX_FORM_TUPLES)) ' Safety stop here
 j = InStr(i, sList, Chr$(0)) ' J -> next null
 CGI_FormTuples(n).key = Mid$(sList, i, j - i) ' Get Key, then pathname
 buf = GetProfile("Form External", CGI_FormTuples(n).key)
 k = InStr(buf, " ") ' Split file & length
 extName = Mid$(buf, 1, k - 1) ' Pathname
 k = k + 1
 extlen = CLng(Mid$(buf, k, Len(buf) - k + 1)) ' Length
 '
 ' Use feature of GET to read content in one call
 '
 Open extName For Binary Access Read As #extFile
```

→

ลิสติงที่ 1 (ต่อ)

```
    CGI_FormTuples(n).value = String$(extlen, " ") ' Breathe in...
    Get #extFile, , CGI_FormTuples(n).value 'GULP!
    Close #extFile
    i = j + 1 ' Bump pointer
    n = n + 1 ' Bump array index
    Loop

    CGI_NumFormTuples = n ' Number of fields decoded
    n = 0 ' Reset counter
    '
    ' Next, the [Form Huge] section. Will this ever get executed?
    '
    sList = GetProfile("Form Huge", "") ' Get key list
    l = Len(sList) ' Length incl. trailing null
    i = 1 ' Start at 1st character
    Do While ((i < l) And (n < MAX_FORM_TUPLES)) ' Safety stop here
    j = InStr(i, sList, Chr$(0)) ' J -> next null
    CGI_HugeTuples(n).key = Mid$(sList, i, j - i) ' Get Key
    buf = GetProfile("Form Huge", CGI_HugeTuples(n).key) ' "offset length"
    k = InStr(buf, " ") ' Delimiter
    CGI_HugeTuples(n).offset = CLng(Mid$(buf, 1, (k - 1)))
    CGI_HugeTuples(n).length = CLng(Mid$(buf, k, (Len(buf) - k + 1)))
    i = j + 1 ' Bump pointer
    n = n + 1 ' Bump array index
    Loop

    CGI_NumHugeTuples = n ' Fill in global count

    n = 0 ' Reset counter
    '
    ' Finally, the [Form File] section.
    '
    sList = GetProfile("Form File", "") ' Get key list
    l = Len(sList) ' Length incl. trailing null
    i = 1 ' Start at 1st character
    Do While ((i < l) And (n < MAX_FILE_TUPLES)) ' Safety stop here
    j = InStr(i, sList, Chr$(0)) ' J -> next null
    CGI_FileTuples(n).key = Mid$(sList, i, j - i) ' Get Key
    buf = GetProfile("Form File", CGI_FileTuples(n).key)
    ParseFileValue buf, CGI_FileTuples(n) ' Complicated, use Sub
    i = j + 1 ' Bump pointer
    n = n + 1 ' Bump array index
    Loop

    CGI_NumFileTuples = n ' Fill in global count

End Sub

'-------------------------------------------
'
' GetProfile() - Get a value or enumerate keys in CGI_Profile file
'
' Get a value given the section and key, or enumerate keys given the
' section name and "" for the key. If enumerating, the list of keys for
' the given section is returned as a null-separated string, with a
' double null at the end.
'
' VB handles this with flair! I couldn't believe my eyes when I tried this.
'-------------------------------------------
Private Function GetProfile(sSection As String, sKey As String) As String
    Dim retLen As Long
    Dim buf As String * ENUM_BUF_SIZE

    If sKey <> "" Then
    retLen = GetPrivateProfileString(sSection, sKey, "", buf, ENUM_BUF_SIZE, CGI_ProfileFile)
    Else
    retLen = GetPrivateProfileString(sSection, 0&, "", buf, ENUM_BUF_SIZE, CGI_ProfileFile)
    End If
    If retLen = 0 Then
    GetProfile = ""
    Else
    GetProfile = Left$(buf, retLen)
    End If

End Function

'-------------------------------------------
```

→

ลิสติงที่ 1 (ต่อ)

```
'
' Get the value of a "small" form field given the key
'
' Signals an error if field does not exist
'
'-------------------------------------
Function GetSmallField(key As String) As String
 Dim i As Integer

 For i = 0 To (CGI_NumFormTuples - 1)
 If CGI_FormTuples(i).key = key Then
 GetSmallField = Trim$(CGI_FormTuples(i).value)
 Exit Function ' ** DONE **
 End If
 Next i
 '
 ' Field does not exist
 '
 Error ERR_NO_FIELD
End Function

'-------------------------------------
'
' InitializeCGI() - Fill in all of the CGI variables, etc.
'
' Read the profile file name from the command line, then fill in
' the CGI globals, the Accept type list and the Extra headers list.
' Then open the input and output files.
'
' Returns True if OK, False if some sort of error. See ReturnError()
' for info on how errors are handled.
'
' NOTE: Assumes that the CGI error handler has been armed with On Error
'-------------------------------------
Sub InitializeCGI()
 Dim sect As String
 Dim argc As Integer
 Static argv(MAX_CMDARGS) As String
 Dim buf As String

 CGI_DebugMode = True ' Initialization errors are very bad

 '
 ' Parse the command line. We need the profile file name (duh!)
 ' and the output file name NOW, so we can return any errors we
 ' trap. The error handler writes to the output file.
 '
 argc = GetArgs(argv())
 CGI_ProfileFile = argv(0)

 sect = "CGI"
 CGI_ServerSoftware = GetProfile(sect, "Server Software")
 CGI_ServerName = GetProfile(sect, "Server Name")
 CGI_RequestProtocol = GetProfile(sect, "Request Protocol")
 CGI_ServerAdmin = GetProfile(sect, "Server Admin")
 CGI_Version = GetProfile(sect, "CGI Version")
 CGI_RequestMethod = GetProfile(sect, "Request Method")
 buf = GetProfile(sect, "Request Keep-Alive") ' Y or N
 If (Left$(buf, 1) = "Y") Then ' Must start with Y
 CGI_RequestKeepAlive = True
 Else
 CGI_RequestKeepAlive = False
 End If
 CGI_LogicalPath = GetProfile(sect, "Logical Path")
 CGI_PhysicalPath = GetProfile(sect, "Physical Path")
 CGI_ExecutablePath = GetProfile(sect, "Executable Path")
 CGI_QueryString = GetProfile(sect, "Query String")
 CGI_RemoteHost = GetProfile(sect, "Remote Host")
 CGI_RemoteAddr = GetProfile(sect, "Remote Address")
 CGI_RequestRange = GetProfile(sect, "Request Range")
 CGI_Referer = GetProfile(sect, "Referer")
 CGI_From = GetProfile(sect, "From")
 CGI_UserAgent = GetProfile(sect, "User Agent")
 CGI_AuthUser = GetProfile(sect, "Authenticated Username")
 CGI_AuthPass = GetProfile(sect, "Authenticated Password")
 CGI_AuthRealm = GetProfile(sect, "Authentication Realm")
 CGI_AuthType = GetProfile(sect, "Authentication Method")
```

→

ลิสติงที่ 1 (ต่อ)

```
    CGI_ContentType = GetProfile(sect, "Content Type")
    buf = GetProfile(sect, "Content Length")
    If buf = "" Then
    CGI_ContentLength = 0
    Else
    CGI_ContentLength = CLng(buf)
    End If
    buf = GetProfile(sect, "Server Port")
    If buf = "" Then
    CGI_ServerPort = -1
    Else
    CGI_ServerPort = CInt(buf)
    End If

    sect = "System"
    CGI_ContentFile = GetProfile(sect, "Content File")
    CGI_OutputFile = GetProfile(sect, "Output File")
    CGI_OutputFN = FreeFile
    Open CGI_OutputFile For Output Access Write As #CGI_OutputFN
    buf = GetProfile(sect, "GMT Offset")
    If buf <> "" Then ' Protect against errors
    CGI_GMTOffset = CVDate(Val(buf) / 86400#) ' Timeserial GMT offset
    Else
    CGI_GMTOffset = 0
    End If
    buf = GetProfile(sect, "Debug Mode") ' Y or N
    If (Left$(buf, 1) = "Y") Then ' Must start with Y
    CGI_DebugMode = True
    Else
    CGI_DebugMode = False
    End If

    GetAcceptTypes ' Enumerate Accept: types into tuples
    GetExtraHeaders ' Enumerate extra headers into tuples
    GetFormTuples ' Decode any POST form input into tuples

End Sub

'-------------------------------------
'
' main() - CGI script back-end main procedure
'
' This is the main() for the VB back end. Note carefully how the error
' handling is set up, and how program cleanup is done. If no command
' line args are present, call Inter_Main() and exit.
'-------------------------------------
Sub Main()
    On Error GoTo ErrorHandler

    If Trim$(Command$) = "" Then ' Interactive start
    Inter_Main ' Call interactive main
    Exit Sub ' Exit the program
    End If

    InitializeCGI ' Create the CGI environment

    '============
    CGI_Main ' Execute the actual "script"
    '============

Cleanup:
    Close #CGI_OutputFN
    Exit Sub ' End the program
'-------
ErrorHandler:
    Select Case Err ' Decode our "user defined" errors
    Case ERR_NO_FIELD:
    ErrorString = "Unknown form field"
    Case Else:
    ErrorString = Error$ ' Must be VB error
    End Select

    ErrorString = ErrorString & " (error #" & Err & ")"
    On Error GoTo 0 ' Prevent recursion
    ErrorHandler (Err) ' Generate HTTP error result
    Resume Cleanup
'-------
```

→

ลิสติงที่ 1 (ต่อ)

```
End Sub

'_____________________________________
'
' Send() - Shortcut for writing to output file
'
'_____________________________________
Sub Send(s As String)
 Print #CGI_OutputFN, s
End Sub

'_____________________________________
'
' SendNoOp() - Tell browser to do nothing.
'
' Most browsers will do nothing. Netscape 1.0N leaves hourglass
' cursor until the mouse is waved around. Enhanced Mosaic 2.0
' oputs up an alert saying "URL leads nowhere". Your results may
' vary...
'
'_____________________________________
Sub SendNoOp()

 Send ("HTTP/1.0 204 No Response")
 Send ("Server: " + CGI_ServerSoftware)
 Send ("")

End Sub

'_____________________________________
'
' WebDate - Return an HTTP/1.0 compliant date/time string
'
' Inputs: t = Local time as VB Variant (e.g., returned by Now())
' Returns: Properly formatted HTTP/1.0 date/time in GMT
'_____________________________________
Function WebDate(dt As Variant) As String
 Dim t As Variant

 t = CVDate(dt - CGI_GMTOffset) ' Convert time to GMT
 WebDate = Format$(t, "ddd dd mmm yyyy hh:mm:ss") & " GMT"

End Function


'_____________________________________
'
' PlusToSpace() - Remove plus-delimiters from HTTP-encoded string
'
'_____________________________________
Public Sub PlusToSpace(s As String)
 Dim i As Integer

 i = 1
 Do While True
 i = InStr(i, s, "+")
 If i = 0 Then Exit Do
 Mid$(s, i) = " "
 Loop

End Sub

'_____________________________________
'
' Unescape() - Convert HTTP-escaped string to normal form
'
'_____________________________________
Public Function Unescape(s As String)
 Dim i As Integer, l As Integer
 Dim c As String

 If InStr(s, "%") = 0 Then ' Catch simple case
 Unescape = s
 Exit Function
 End If

 l = Len(s)
```

→

ติงที่ 1 (ต่อ)

```
Unescape = ""
For i = 1 To l
c = Mid$(s, i, 1) ' Next character
If c = "%" Then
If Mid$(s, i + 1, 1) = "%" Then
c = "%"
i = i + 1 ' Loop increments too
Else
c = x2c(Mid$(s, i + 1, 2))
i = i + 2 ' Loop increments too
End If
End If
Unescape = Unescape & c
Next i

End Function

'---------------------------------------
'
' x2c() - Convert hex-escaped character to ASCII
'
'---------------------------------------
Private Function x2c(s As String) As String
 Dim t As String

 t = "&H" & s
 x2c = Chr$(CInt(t))

End Function

Private Sub ParseFileValue(buf As String, ByRef t As FileTuple)
 Dim i, j, k, l As Integer

 l = Len(buf)

 i = InStr(buf, " ") ' First delimiter
 t.file = Mid$(buf, 1, (i - 1)) ' [file]
 t.file = Mid$(t.file, 2, Len(t.file) - 2) ' file

 j = InStr((i + 1), buf, " ") ' Next delimiter
 t.length = CLng(Mid$(buf, (i + 1), (j - i - 1)))
 i = j

 j = InStr((i + 1), buf, " ") ' Next delimiter
 t.type = Mid$(buf, (i + 1), (j - i - 1))
 i = j

 j = InStr((i + 1), buf, " ") ' Next delimiter
 t.encoding = Mid$(buf, (i + 1), (j - i - 1))
 i = j

 t.name = Mid$(buf, (i + 1), (l - i - 1)) ' [name]
 t.name = Mid$(t.name, 2, Len(t.name) - 1) ' name

End Sub

'---------------------------------------
'
' FindExtraHeader() - Get the text from an "extra" header
'
' Given the extra header's name, return the stuff after the ":"
' or an empty string if not there.
'---------------------------------------
Public Function FindExtraHeader(key As String) As String
 Dim i As Integer

 For i = 0 To (CGI_NumExtraHeaders - 1)
 If CGI_ExtraHeaders(i).key = key Then
 FindExtraHeader = Trim$(CGI_ExtraHeaders(i).value)
 Exit Function ' ** DONE **
 End If
 Next i
 '
 ' Not present, return empty string
 '
 FindExtraHeader = ""
End Function
```

โปรแกรม Main นี้เป็นส่วนแรกสุดที่โปรแกรมจะเริ่มทำงาน เริ่มแรกจากตรวจสอบว่าถูกเรียกจากเวบเซอร์ฟเวอร์หรือเปล่า ใน Visual Basic นี้จะใช้ฟังก์ชัน Command เป็นการรับ argument จาก Command Line เช่น การเรียกใช้ฟังก์ชันนี้ก็เพื่อรับชื่อไฟล์ Temporary มาเก็บเอาไว้ จะเห็นว่ามีการตรวจสอบในบรรทัด If Trim$(Command$) = "" Then ตรงจุดนี้เป็นการระบุว่าถ้ามีการเรียกใช้โปรแกรมโดยที่ไม่มี Argument ส่งมาให้ ก็ให้ถือว่ามีข้อผิดพลาดและให้แจ้ง Error ข้อผิดพลาดซึ่งจะถูกเรียกโดย Sub Inter_Mail ซึ่งคุณจะต้องสร้างขึ้นมาด้วยทุกครั้ง ไม่ว่าคุณจะมีข้อความชี้แจงผิดพลาดอะไรหรือไม่ก็ตาม แต่ถ้าคุณไม่อยากเสียเวลาสร้างทุกครั้งผมแนะนำว่า คุณควรจะสร้าง sub นี้ไว้ใน CGI32.BAS เลย และให้แจ้งข้อความ Error ที่เป็นมาตรฐานออกมาก็แค่นั้น จุดต่อไปก็คือ ฟังก์ชัน InitializeCGI (ดูตัวอย่างที่ 2)

ตัวอย่างที่ 2

```
Sub Main()
 On Error GoTo ErrorHandler

 If Trim$(Command$) = "" Then ' Interactive start
 Inter_Main ' Call interactive main
 Exit Sub ' Exit the program
 End If

 InitializeCGI ' Create the CGI environment

 '============
 CGI_Main ' Execute the actual "script"
 '============

Cleanup:
 Close #CGI_OutputFN
 Exit Sub ' End the program
'-------
ErrorHandler:
 Select Case Err ' Decode our "user defined" errors
 Case ERR_NO_FIELD:
 ErrorString = "Unknown form field"
 Case Else:
 ErrorString = Error$ ' Must be VB error
 End Select

 ErrorString = ErrorString & " (error #" & Err & ")"
 On Error GoTo 0 ' Prevent recursion
 ErrorHandler (Err) ' Generate HTTP error result
 Resume Cleanup
'-------
End Sub
```

ฟังก์ชัน InitializeCGI

ถือว่าเป็นฟังก์ชันที่มีความสำคัญมาก เนื่องจากว่าฟังก์ชันนี้จะทำหน้าที่อ่านค่าต่าง ๆ ขึ้นมาจาก CGI Profile ซึ่งถ้าใช้โปรแกรม WinSite Web Server ตัวโปรแกรมจะสร้างให้เป็นนามสกุล INI แต่ถ้าคุณใช้โปรแกรมเวบเซอร์ฟเวอร์ตัวอื่น ๆ มันอาจจะไม่สร้างไว้เป็นนามสกุล .INI อย่างผมลองใช้กับ OmniHTTP นี้มันจะสร้างมาเป็น .TMP ให้ แต่ทั้งนี้ทั้งนั้นมันก็ไม่ใช่ส่วนที่เสียหายอะไรเพราะการเรียกใช้โปรแกรม CGI-WIN ผ่านทางเวบเซอร์ฟเวอร์นี้ โปรแกรมเวบเซอร์ฟเวอร์ก็ต้องมีหน้าที่บอกชื่อไฟล์ให้อยู่แล้ว อย่างที่ผมได้ยกตัวอย่างของ CGI Profile คุณจะเห็นว่าจะถูกแบ่งออกเป็น Section ต่าง ๆ จึงต้องใช้ฟังก์ชัน GetPrivateProfileString ซึ่งเป็น API Function ของวินโดวส์มาใช้ ที่ต้องใช้ฟังก์ชันนี้ก็เพราะว่ามันสะดวกในการดึงค่าต่าง ๆ ขึ้นมา อย่างแรกเลยคือคุณไม่ต้องมาเขียนโปรแกรมอ่านค่าเอง (อย่างที่ผมทำกับ Clipper) ถ้าคุณไล่ดูจากโปรแกรม CGI32.BAS คุณจะเห็นว่ามีการ Declear ฟังก์ชันขึ้นใช้งาน ส่วนรายละเอียดการเรียกใช้ยังไงก็ลองไปไล่ดูเองนะครับไม่ยากเท่าไร ฟังก์ชัน InitialzineCGI นี้จะทำการอ่านฟิลด์ข้อมูลในฟอร์มให้คุณเสร็จสรรพเรียบร้อยทุกอย่าง สำหรับชื่อตัวแปร Env ที่คุณต้องการเรียกใช้คุณสามารถอ้างชื่อตามตารางที่ 2 ได้เลยครับ

ตารางที่ 2

ชื่อตัวแปร	ความหมาย
CGI Version	The version of CGI spoken by the server
Request Protocol	The server's info protocol (e.g. HTTP/1.0)
Request Method	The method specified in the request (e.g., "GET")
Request Keep-Alive	If the client requested connection re-use (Yes/No)
Executable Path	Physical pathname of the back-end (this program)
Logical Path	Extra path info in logical space
Physical Path	Extra path info in local physical space
Query String	String following the "?" in the request URL
Content Type	MIME content type of info supplied with request
Content Length	Length, bytes, of info supplied with request
Request Range	Byte-range specfication received with request
Server Software	Version/revision of the info (HTTP) server
Server Name	Server's network hostname (or alias from config)
Server Port	Server's network port number
Server Admin	E-Mail address of server's admin. (config)
Referer	URL of referring document
From	E-Mail of client user (rarely seen)
User Agent	String describing client/browser software/version
Remote Host	Remote client's network hostname
Remote Address	Remote client's network address
Authenticated Username	Username if present in request
Authenticated Password	Password if present in request
Authentication Method	Method used for authentication (e.g., "Basic")
Authentication Realm	Name of realm for users/groups

ตารางที่ 3 แสดงชื่อฟังก์ชันที่คุณต้องเรียกใช้

ชื่อฟังก์ชัน	ความหมาย
CGI_Main()	คุณต้องสร้างเองทุกครั้ง
Inter_Main()	เป็น Function ตรวจสอบว่ามีการเรียกจาก Web Server หรือเปล่า ซึ่งก็อีกนั่นแหละคุณต้องสร้างเอาไว้ด้วย
GetSmallField(String)	อ่านข้อมูลฟิลด์ที่อยู่ในฟอร์ม วิธีใช้ให้ใส่ชื่อฟิลด์ที่ต้องการอ่านข้อมูล
Send(String)	เอาไว้ส่งข้อความไปให้ Web Server

สติงที่ 2 โปรแกรมตัวอย่าง VBCGI.BAS

```
Attribute VB_Name = "Module1"
Sub CGI_Main()
 Call StartDocument
End Sub

Sub StartDocument()
 Dim mydata As Database
 Dim rcorder As Recordset
 Set mydata = OpenDatabase("Sale.mdb") ' เปิด Database
 Set rcordset = mydata.OpenRecordset("order") ' เปิด Table

 rcordset.MoveFirst

 Send ("Content-type: text/html") ' ส่ง Content Type ไปให้กับ Web
Server
 Send ("X-CGI-Program: VB4 CGI Tester 1.2")
 Send ("")
 Send ("<HTML><HEAD><TITLE>Windows CGI Test Results</TITLE></
HEAD>")
 Send ("<BODY><H1>Windows CGI Test Results</H1>")
 Send ("Program version: 1.3 (32-bit) (16-Nov-95)<BR>")
 Send ("Server: " & CGI_ServerSoftware & "<BR>")
 Send ("Selector: <CODE>" & sel & "</CODE><P>")
 Send ("<BR>")
 Send ("Customer Name : ") ' แสดงข้อมูล ชื่อและนามสกุล
 Send (rcordset.Fields("Name") & " " & rcordset.Fields("Lname"))
 Send ("<BR>")
 Send ("<HR>")
nd Sub


ub Inter_Main()
MsgBox "This is a CGI program for the Web Server."
' Exit Sub will end the program
nd Sub
```

คราวนี้ก็มาถึงสุดท้ายแล้ว คือตัวอย่างโปรแกรม VGCGI.BAS เป็นโปรแกรมที่ผมได้เขียนขึ้นมาเป็นตัวอย่างเล็ก ๆ หน้าที่ของ CGI รแกรมตัวนี้คือ เปิด Database ขึ้นมา และเปิดเทเบิล Order ขึ้นมา ะแสดงชื่อลูกค้า ไม่มีอะไรซับซ้อน โปรแกรมตัวนี้แสดงให้เห็นว่าคุณ มารถที่จะใช้ดาต้าเบสของ MS-Access ในการเก็บข้อมูลแสดงข้อมูล สังเกตว่าจะต้องสร้าง Sub CGI_Main ด้วยเสมอ เพราะจะต้องมีการ ยกใช้ที่นี่เป็นจุดเริ่มต่อจาก Main() ใน CGI32.BAS ครับส่วน er_Main() ผมได้แสดงข้อความผิดพลาดเล็ก ๆ เอาไว้ ถ้าคุณดูจาก โค้ดของโปรแกรมคุณจะพบว่าภายในคำสั่ง SEND นี้ จะเป็นการส่งข้อความที่อยู่ในรูปของ HTML ออกไป ซึ่งเมื่อคุณได้สั่ง SEND แล้ว โปรแกรมจะเขียนลงไปในไฟล์ Output ตามที่ผมได้กล่าวถึงไว้ และเวบเซอร์ฟเวอร์จะเป็นตัวอ่านข้อมูลนี้และส่งให้กับเวบเบราเซอร์อีกทีและอีกจุดก็คือ Send("Content Type: text/html") ตรงส่วนนี้เป็น Content Header ซึ่งคุณจะต้องระบุด้วยนะครับ และควรเว้นบรรทัดเอาต่อจากบรรทัดนี้ 2 บรรทัด ตรงจุดนี้เป็นส่วนที่สำคัญนะครับ เพราะจะเป็นการบอกให้เวบเซอร์ฟเวอร์รู้ว่าข้อมูลที่ส่งเข้ามานี้เป็นอะไร ถ้าคุณเปิดดูใน IE ในส่วนของ Option/Program/Viewer/File Type คุณจะเห็น text/html ไปเปิดดูเอาแล้วจะเข้าใจครับ

ส่งท้าย

สำหรับผู้ที่ใช้ VB 3.0 หรือ VB 4.0 เวอร์ชัน 16 บิต คุณต้องใช้ CGI.BAS แทนนะครับ รับรองว่าทำงานได้อย่างแน่นอน